REMARKS

Claims 1, 2, 10, 11, 14, 16, 17, 20, 26, 31 and 35 have been amended. Claims 12, 27 and 36 have been cancelled. Therefore, claims 1-11, 13-26, 28-35 and 37 are pending in the application. Reconsideration is respectfully requested in light of the following remarks.

Section 103(a) Rejections:

The Office Action rejected claims 1-4, 6-9, 17, 19, 20, 22-25 and 31-34 under 35 U.S.C. § 103(a) as being unpatentable over Montero et al. (U.S. Publication 2002/0143958) (hereinafter "Montero") in view of Bennett (U.S. Patent 5,734,909). Applicants respectfully traverse this rejection for at least the reasons below.

Regarding claim 1, Montero in view of Bennet fails to teach or suggest a distributed store configured to send a lock token to the process, wherein only processes that have received a lock token can access the primary state. Neither Montero nor Bennet teach anything regarding lock tokens or about a distributed store sending a lock token to a process as part of providing locked access to the primary state of session data. Montero, as admitted by the Examiner, fails to teach or suggest any locking mechanism. Bennet teaches a system in which server processes gain locked access to server resources, but Bennet's system does not include the use of lock tokens. Instead, Bennet teaches that a data structure on the server is used to record which client (and therefore which server process) holds the lock on a particular resource (see, Bennet, column 7, lines 22-35). Thus, the combination of Montero and Bennet fails to teach or suggest a distributed store configured to send a lock token to the process, wherein only processes that have received a lock token can access the primary state. For at least the reasons above, the rejection of claim 1 is not supported by the prior art and removal thereof is respectfully requested. Similar remarks apply to claims 17, 20 and 31.

Regarding claim 2, Montero in view of Bennet fails to teach or suggest wherein after a process has completed a current access of the primary state, the process is configured to hold locked access until after receiving a request to release the locked access. The Examiner cites (regarding claim 12) column 1, lines 58-65 of Bennet. However, this passage of Bennet only describes how Bennet's system allows multiple clients to have read access to the same resource, but does not allow read access to resources already having an exclusive write lock. The Examiner also cites column 3, line 54 to column 4, line 14 of Bennet. This passage of Bennet describes how in Bennet's system when a client request requires locked access to a resource already locked by another processes, the client request is queued until the resource is available. The Examiner additionally cites lines 22-46 of column 7 in Bennet that describes how when a process releases a locked resource and there is another client request waiting for locked access to the same resource, the process releasing the resources gives it to the waiting client. The Examiner further cites column 8, lines 14-35 where Bennet describes a client requests the unlocking of a resource held by the client and the relocking of that resource for another client waiting for that resource. However, none of the cited passages refers to a process configured to hold locked access, after completing a current access, until after receiving a request to release the locked access. Instead, as described at the Examiner's cited passages, Bennet teaches that a second client request requiring access to an already locked resource is queued until the locked resource is released. Nowhere does Bennet mention requesting a process to release the locked access.

Montero is not relied upon by the Examiner to teach anything regarding locking resources. Thus, the combination of Montero and Bennet fails to teach or suggest after the process has completed a current access of the primary state, the process is configured to hold locked access until after receiving a request to release the locked access. Thus, for at least the reasons above, the rejection of claim 2 is not supported by the prior art and removal thereof is respectfully requested.

Regarding claim 3, Montero in view of Bennet fails to teach or suggest wherein the distributed store is configured to request the process to release the locked access, wherein the process is configured to release the locked access in response to the request. The Examiner cites the same passages of Bennet cited in the rejection of claim 2, discussed above. None of the cited passages teaches or suggests the distributed store requesting a process to release the locked access. Instead, as noted above, Bennet teaches that client request requiring access to an already locked resource are queued until the resource is available. For a discussion of the cited passages, please see the above discussion of claim 2. As Montero fails to teach anything regarding any locking mechanism, Montero fails to over this deficiency of Bennet. Thus, the combination of Montero in view of Bennet fails to teach or suggest wherein the distributed store is configured to request the process to release the locked access, wherein the process is configured to release the locked access in response to the request. Therefore, for at least reasons above, the rejection of claim 3 is not supported by the cited prior art and removal thereof is respectfully requested. Similar remarks apply to claims 22 and 32.

The Office Action rejected claims 5, 10-16, 18, 21, 26-30 and 35-37 under 35 U.S.C. § 103(a) as being unpatentable over Montero in view of Bennett, and further in view of Bender et al. (U.S. Publication 2003/0163494) (hereinafter "Bender"). Applicants respectfully traverse this rejection for at least the reasons below.

Regarding claim 10, Montero in view of Bennet in further view of Bender fails to teach or suggest a distributed store configured to request the process to release the locked access, wherein the process in configured to release the locked access in response to the request. Montero does not teach any sort of locking mechanism, as admitted by the Examiner. Bennet teaches a system in which client requests requiring locked access to a resource already locked are queued until the locked resource is available (Bennet, column 3, line 62 – column 4, line 14). Bender teaches a system in which processes and threads requiring locked access to already locked resources must wait until the process holding locked access to the resource has finished with the resource. Neither Benner nor Bender teach a distributed store configured to request the process to release the locked access.

The Examiner cites several passages of Bennet (column 1, lines 58-65; column 3, line 54 – column 4, line 14; column 7, lines 22-46; and column 8, lines 14-35) describing how in Bennet's system client requests requiring locked access to a shared resource are queued until a server processing currently holding a lock on the resource is finished with, and releases, the resource, as described above regarding claim 2. Specifically, column 1, lines 58-65 of Bennet only describes how Bennet's system allows multiple clients to have read access to the same resource, but does not allow read access to resources already having an exclusive write lock. Column 3, line 54 to column 4, line 14 of Bennet describes how in Bennet's system when a client request requires locked access to a resource already locked by another processes, the client request is queued until the resource is available. Lines 22-46 of column 7 in Bennet describe how when a process releases a locked resource and there is another client request waiting for locked access to the same resource, the process releasing the resources gives it to the waiting client. Column 8, lines 14-35 of Bennet describes a client requests the unlocking of a resource held by the client and the relocking of that resource for another client waiting for that resource. However, None of the cited passages mentions anything regarding <u>the distributed store requesting</u> the process (holding locked access to a resource) to release the locked access, wherein the process is configured to release the locked access <u>in response to</u> the request.

The Examiner does not rely upon bender, nor does Bender teach or suggest a distributed store configured to request the process to release the locked access. Since Montero, Bennet and Bender all fail to teach or suggest wherein the distributed store is configured to request the process to release the locked access, wherein the process is configured to release the locked access in response to said request, the combination of Montero, Bennet and Bender also fails to teach or suggest such functionality. For at least the reasons below, the rejection of claim 10 is not supported by the prior art and removal thereof is respectfully requested. Similar remarks apply to claims 26 and 35.

Regarding claim 11, Montero in view of Bennet further fails to teach or suggest after the process has completed a current access of the primary state, the process is configured to hold locked access until after receiving a request to release the locked access. The Examiner cites (regarding former claim 12) column 1, lines 58-65 of Bennet. However, this passage of Bennet only describes how Bennet's system allows multiple clients to have read access to the same resource, but does not allow read access to resources already having an exclusive write lock. The Examiner also cites column 3, line 54 to column 4, line 14 of Bennet. This passage of Bennet describes how in Bennet's system when a client request requires locked access to a resource already locked by another processes, the client request is queued until the resource is available. The Examiner additionally cites lines 22-46 of column 7 in Bennet that describes how when a process releases a locked resource and there is another client request waiting for locked access to the same resource, the process releasing the resources gives it to the waiting client. The Examiner further cites column 8, lines 14-35 where Bennet describes a client requests the unlocking of a resource held by the client and the relocking of that resource for another client waiting for that resource. However, none of the cited passages refers to a process configured to hold locked access, after completing a current access, until after receiving a request to release the locked access. Instead, as described at the Examiner's cited passages, Bennet teaches that a second client request requiring access to an already locked resource is queued until the locked resource is released. Nowhere does Bennet mention requesting a process to release the locked access.

Neither Montero nor Bender are not relied upon by the Examiner to teach, nor do they teach, anything regarding a process holding locked access until receiving a request to release the locked access. Thus, the combination of Montero, Bennet and Bender fails to teach or suggest after the process has completed a current access of the primary state, the process is configured to hold locked access until after receiving a request to release the locked access. Thus, for at least the reasons above, the rejection of claim 11 is not supported by the prior art and removal thereof is respectfully requested.

Applicants also asserts that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the rejection has been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time.

CONCLUSION

Applicants submit the application is in condition for allowance, and notice to that effect is respectfully requested.

If any extension of time (under 37 C.F.R. § 1.136) is necessary to prevent the above referenced application from becoming abandoned, Applicants hereby petition for such extension. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-11700/RCK.

Also enclosed herewith are the following items:

☒ Return Receipt Postcard

□ Petition for Extension of Time

□ Notice of Change of Address

□ Fee Authorization Form authorizing a deposit account debit in the amount of $ for fees (   ).

□ Other:

Respectfully submitted,

Robert C. Kowert
Reg. No. 39,255
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: July 15, 2005